

Ne-klauzalni SAT resavači

Automatsko rezonovanje

Milan Kocić 1037/2021

Uvod

Većina SAT rešavača radi sa formulama u normalnim formama. Mnoge probleme je lakše opisati opštom iskaznom formulom, da bi se ispitala zadovoljivost takvih formula standardno se prevedi u konjuktivnu normalnu formu, što se može izbeći. Jedna od prednosti izbegavanja transformacije u normalne forme je to što formula čiju zadovoljivost ispitujemo direktno prikazuje strukturu problema. Takođe u toku transformacije se javljaju i nove promenljive, pa se i veličina ulaza povećava.

Metoda

Formula se čuva u DAG formatu, odnosno kao usmereni aciklični graf, gde se svaka od podformula čuva samo jednom. Transformacija se vrši tako što pamtimo koje podformule su se do sada pojavile, ukoliko se neka pojavi ponovo, koristi se stara. Algoritam radi nad formulom koja je ovako predstavljena.

Na početku je potrebno postaviti celoj formuli tačnost T, i svim podformulama proslediti posledice te odluke, ukoliko se među posledicama pojavi neki konflikt formula nije zadovoljiva. Inače se bira jedna od podformuli čija se tačnost još uvek ne zna, izabere se neka tačnost te podformule, i proslede se sve posledice te odluke. Ukoliko se pojavi konflikt, a imali smo izbora pre toga, uzima se poslednji izbor, sve posledice tog izbora se vraćaju na stanje pre izbora, podformuli se menja tačnost, ovo više nije izbor, već smo primorani da postavimo novu tačnost, samim tim i ne pamtimo ovo kao izbor. Ukoliko nastane konflikt, a nije bilo izbora koji se može promeniti, formula nije zadovoljiva. Ukoliko je svim podformulama dodeljena neka tačnost i nije došlo do konflikta, našli smo valuaciju koja zadovoljava traženu formulu.

Implementacija

Za računanje DAG reprezentacije se koristi mapa koja proverava da li je neka formula već definisana do sada, ukoliko jeste koristi se ta stara formula, a visak memorije se oslobadja. Implementirana je klasa dagcvor kao i podklase koje predstavljaju svaki od logičkih veznika. Konjunkcija i disjunkcija su predstavljeni kao n-arni veznici, dok su ostali veznici predstavljeni standardno.

Uz sveki čvor u DAG reprezentaciji pored standardnih informacija o podformulama čvora, čuvaju se i informacije o svim formulama koje su nadformule datog čvora, informacija o tačnosti čvora(T,NT,Nepoznato,Nebitno), takodje čuva se i vektor sa informacijama o posledica izbora tačnosti neke formule.

Svaka od klasa koje nasledjuju dagcvor ima implementiran metod propagate, koji je u suštini i glavni deo implementacije rešavača. Njemu se prosledjuje tačnost koju dobija čvor, čvor zbog čijeg izbora se menja tačnost, i informacija da li je informacija stigla odozgo ili odozdo u DAG reprezentaciji, metod vraća true/false u zavisnosti od uspeha propagacije. Kada neki čvor preko metoda propagate dobije informacije koje su mu dovoljne da promeni tačnost on o tome kroz metod propagate obaveštava i roditelje i potomke. Ukoliko se informacija šalje potomcima, ona je sigurna i čvor koji je dobio informaciju postavlja traženu tačnost ili prijavljuje konflikt. Medjutim, ukoliko informaciju saljemo roditelju, ona je samo informativnog tipa i roditelj u zavisnosti od slučaja bira šta da radi. Na primeru konjukcije, ako roditelj pozove propagate sa tačnošću T, moguća su tri slučaja, jedan u kome je tačnost konjukcije NT(prijavljuje se konflikt), drugi gde je tačnost T tada je već odradjen sav posao i samo vraćamo tačno, i treći kada konjukcija ima nedefinisanu vrednost. U tom slučaju se tačnost postavlja na T, salje se informacija nadformulama da se promenila tačnost, i šalje se informacija potomcima da svi moraju postaviti svoju tačnost na T jer je konjukcija tačna samo kada su svi njeni potomci tačni. Slično se dešava i u ostalim slučajevima.

Problem se rešava pozivom metoda resi klase dagcvor, koji celoj formuli postavi tačnost na T, a zatim sve dok ima formula kojoj nije dodeljena neka tačnost, poziva se metod propagate, ukoliko on vrati tačno, formula se dodaje u vektor izabrao, inače se proba druga tačnost, ukoliko ni taj poziv metoda propagate ne vrati tačno, uzima se poslednji element vektora izabrao, vraća se sve na stanje pre njegove propagacije, brise se iz vektora izabrao, i propagira se sa odgovarajućom tačnošću. Ukoliko je vektor izabrao prazan u trenutku kada bi trebalo uzeti formulu sa kraja(probali smo sve izbore koje smo mogli napraviti), formula nije zadovoljiva.

Prevodjenje i pokretanje

Program se prevodi narednom komandom: `g++ main.cpp lexer.cpp parser.cpp dag.cpp`

A zatim se pokreće sa: `./a.out`

Na standardni ulaz uneti formulu čiju zadovoljivost je potrebno ispitati (\sim negacija, $\&$ konjukcija, $|$ disjunkcija, \Rightarrow implikacija, \Leftrightarrow ekvivalencija).

U fajlu dag.cpp se nalazi promenljiva `stampaj_debug` koju je moguće promeniti na tačno za vise detalja o postupku izvršavanja programa.